

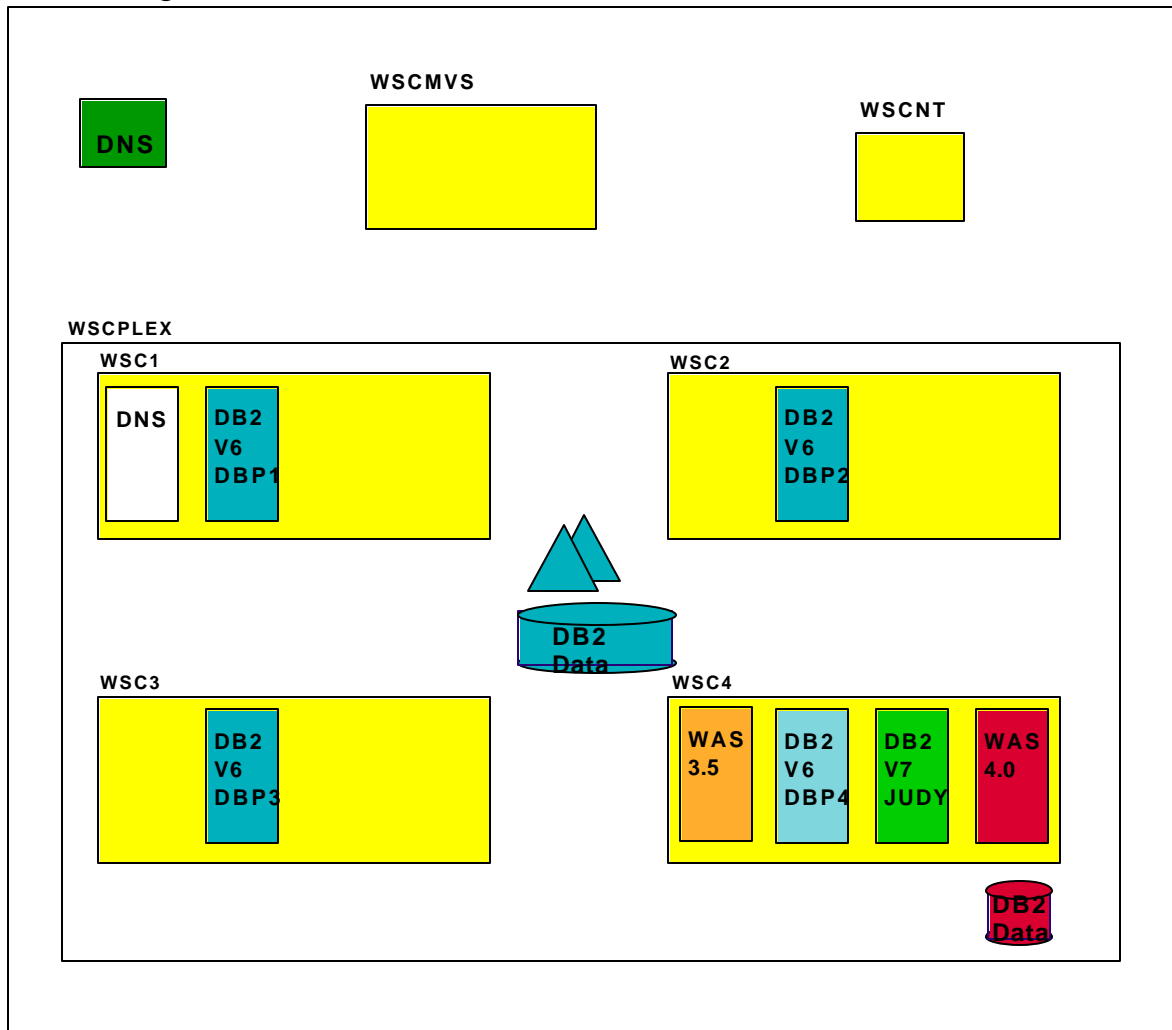
The WAS 4.0 DB2 V7 Conundrum

Mike Cox
Judy Ruby-Brown

The Washington System Center has the same dilemma as a number of other DB2 customers. Namely, all the production applications using DB2 are connected to a DB2 V6 subsystem and there is interest in implementing WAS on OS/390 as quickly as possible; but there is the DB2 V7 requirement for WAS 4.0.

The following diagram is a representation of the WSC environment.

washington.ibm.com domain



The WSCPLEX contains a 4-way DB2 data sharing environment. WSC4 was chosen as the target OS/390 image for the WAS 4.0 environment. Because of the WAS 4.0 dependency on DB2 V7, there were three DB2 options we could have chosen:

1. Migrate all 4 members of the data sharing group to DB2 V7 prior to installing WAS and use, DBP4 as the local instance

THE WAS 4.0 DB2 V7 Conundrum

2. Migrate DBP4 subsystem to DB2 V7 and run a mixed level group with a DB2 V7 catalog/directory, and use DBP4 as the local instance.
3. Install a DB2 V7 subsystem separate from the DB2 V6 data sharing group and use DB2's Distributed Data Facility (DDF) to access production data.

The decision was made to install WAS 4.0 on the WSC4 system using option 3, rather than migrate the production data to DB2 V7 at this time.

Note: If your installation does not use DB2 data sharing, this solution is applicable. Data sharing is not a requirement for the solution, but does affect the WSC implementation

TCPIP Information:

The WSCPLEX is a small part of the washington.ibm.com domain. The TCPIP configuration was set up to using the WLM-DNS-sysplex support introduced in OS/390 Release 4. The DB2 data sharing configuration is TCPIP and SNA enabled for remote access. However, for the purposes of this example we will show the TCPIP specific information.

The domain name server on WSC1 is the authoritative DNS for all applications running in the WSCPLEX sysplex. In particular it is responsible for resolving the names which are dynamically registered by subsystems (or application servers) such as DB2, FTP, telnet, WebSphere 4.0, etc. This requires the 'real' washington.ibm.com DNS to have CNAME entries which causes name resolution request to be forwarded to the DNS on WSC1 where it can be properly resolved.

This configuration allows us a great deal of dynamics. Members of the sysplex may be absent, yet FTPing to WSCFTP will succeed if at least one member is up. Similarly, we can handle a system or subsystem failure gracefully as the DNS on WSC1 (or its backup on WSC2) will be aware of which members are active and where. It is not important for this discussion how the generic ipname for the DB2 V6 data sharing group gets resolved, so long as it intelligently routes requests to the correct system.

Shared HFS

As the WSCPLEX is a parallel sysplex, shared HFS has been implemented as well. To overcome the limitation of mounting a single file system at different mount points (caused by different \$VERSION settings) in this environment, all program products which would have had file systems mounted in /usr/lpp/... are actually mounted off a single mount point in the root file system. Thus /usr/lpp/db2/db2610 is actually a symlink to /shared/db2610/db2610 and the hfs containing the db2610 code (i.e., DSN610.DSNHFS) is mounted at /shared/db2610. This allows us to run multiple levels of any program product throughout the sysplex with only one instance of the file system.

DB2 V6 Information:

There are 4 members of the DB2 data sharing group, whose DB2 Group Name is WSCDBP0. Here is the relevant information for each member. Each member of the group must listen on the same port. In this case it is the SQL well known port 446. It is this data we want to access through WAS 4.0.

DBP1

```
LOCATION WSCDBP0
LU      USIBMWZ.WSCDB2A1
GENERICLU USIBMWZ.WSCDBP0
DOMAIN  wscdbp0.wscplex.washington.ibm.com
TCPPOPT 446
RESPORT 5021
```

THE WAS 4.0 DB2 V7 Conundrum

DBP2

```
LOCATION WSCDBP0
LU     USIBMWZ.WSCDB2B1
GENERICLU USIBMWZ.WSCDBP0
DOMAIN  wscdbp0.wscplex.washington.ibm.com
TCPPOINT 446
RESPORT 5022
```

DBP3

```
LOCATION WSCDBP0
LU     USIBMWZ.WSCDB2C1
GENERICLU USIBMWZ.WSCDBP0
DOMAIN  wscdbp0.wscplex.washington.ibm.com
TCPPOINT 446
RESPORT 5023
```

DBP4

```
LOCATION WSCDBP0
LU     USIBMWZ.WSCDB2D1
GENERICLU USIBMWZ.WSCDBP0
DOMAIN  wscdbp0.wscplex.washington.ibm.com
TCPPOINT 446
RESPORT 5024
```

The WSCPLEX DNS 'knows' about all the active DB2 V6 subsystems accepting incoming DDF requests by both the specific and generic names. Requests to resolve the name 'wscdbp0.wscplex.washington.ibm.com' from within the washington.ibm.com domain are handled by the DNS in the wscplex rather than the main DNS for the entire domain.

DB2 V7 Information:

The DB2 configuration/implementation of this subsystem is quite minimal. It exists to support the WAS 4.0 configuration data, the stateful session bean data, the DB2 tables for the Installation Verification Program EJBs and to be a conduit to the DB2 V6 system. Notice this DB2 subsystem does not listen on port 446, because the DBP4 member of the DB2 V6 data sharing group WSCDBP0 uses it.

JUDY

```
LOCATION WSCJUDY
LU     USIBMWZ.WSCDJUDY
GENERICLU -NONE
DOMAIN  wsc4.wscplex.washington.ibm.com
TCPPOINT 5446
RESPORT 5447
```

Since our DB2 V6 product libraries are in the linklist, we must STEPLIB to our DB2 V7 libraries on WSC4.

Accessing the DB2 V6 data via the DB2 V7 subsystem using JDBC.

The gist of the problem is this:

We want BMP and CMP EJBs residing in the WAS 4.0 environment to be backed by data in the production DB2 V6 system. Additionally, we want to continue to allow JDBC access to the production DB2 V6 subsystems from programs running on WSC1, WSC2, WSC3 and WSC4.

THE WAS 4.0 DB2 V7 Conundrum

During the installation of the DB2 V6 subsystem, and the DB2 V7 subsystem, JDBC was enabled for local access. In each instance, the authority to execute the DSNJDBC plan and DSNJDBC.* packages was given to PUBLIC. Then infrastructure was modified to allow JDBC access of data in the DB2 V6 system via the DB2 V7 system

To accomplish this goal we did the following:

1. Set up the Communication Data Base (CDB) in the V7 subsystem.
2. Set up the necessary RACF profiles to control access and set the appropriate DB2 DSNZPARM values to enforce our security rules.
3. Bound the JDBC relevant DBRMs from DB2 V7 into the DB2 V6 subsystem.
4. Tested JDBC access to wscdbp0 from WSC4 using the DB2 V7 subsystem.
5. Configured DataSources in WAS for EJBs to use WSCDBP0 resident data.
6. Granted all necessary DB2 authorities access the data on the V6 DB2 system

Details of how this was accomplished follow:

Step 1: Set up the Communication Data Base (CDB) in the V7 subsystem

This is the CDB information for the JUDY ssid, representing the *server* configuration:

```
INSERT INTO SYSIBM.LOCATIONS (LOCATION, LINKNAME, PORT)
VALUES ('WSCDBP0', 'WSCDBP0', '446');
INSERT INTO SYSIBM.IPNAMES (LINKNAME, SECURITY_OUT, IPADDR)
VALUES ('WSCDBP0', 'R', 'WSCDBP0.WSCPLEX.WASHINGTON.IBM.COM');
```

The LINKNAME value is the generic LUNAME. In our configuration, it has the same name as the DB2 Location Name. The generic LU is used to generate the application name for the RACF PassTicket. Since PassTickets are enabled in our environment, we specified 'R'.

We run the sample on the JUDY ssid as the DB2 requester and access data which exists on the DB2 V6 data sharing group as server. The JUDY CDB is only populated for servers which can be accessed by JUDY.

Step2: Set up RACF PassTicket Support

To set up RACF to generate PassTickets, define and activate a RACF PassTicket data class (PTKTDATA). This class must contain a RACF profile for each remote DB2 subsystem to which you send requests.

1. Activate the RACF PTKTDATA class by issuing the following RACF commands:

```
SETROPTS CLASSACT(PTKTDATA)
SETROPTS RACLIST(PTKTDATA)
```

2. Define profiles for the remote systems by entering the name of each remote system as it appears in the LINKNAME column of table SYSIBM.LOCATIONS in the RACF PTKTDATA class:

```
RDEFINE PTKTDATA wscdbp0 SSIGNON(KEYMASKED(E001193519561977))
RALTER WSCDBP0 APPLDATA('NO REPLAY PROTECTION')
```

3. Refresh RACF PTKTDATA definition:

```
SETROPTS RACLIST(PTKTDATA) REFRESH
```

THE WAS 4.0 DB2 V7 Conundrum

The reason to specify APPLDATA('NO REPLAY PROTECTION') is to deal with the fact that requests being sent to the wscdbp0 system are occurring faster than one per second. This causes non-unique passtickets to be generated by RACF on the sending system, which are rejected on the target system.

The keymask data is 16 hexadecimal characters of your choice. There is nothing significant about our chosen values.

DSNZPARM changes:

We set the DB2 DSNZPARM value TCPALVER=NO on each of the members of the DB2 V6 data sharing group (each DB2 has its own DSNZPARM control block, but this value should be the same for all). This was done meet our security standards.

The DB2 Installation Panel is DSN TIP5 and the parameter is TCP/IP ALREADY VERIFIED. It means that the DB2 V6 servers receive both user ID and PassTicket from any requesters which come through TCP/IP. This is the default.

Step 3: Bind the JDBC relevant DBRMs from DB2 V7 into the DB2 V6 subsystem.

Once the CDB was established, the DBRMs were bound to the DB2 V6 subsystem remotely from the DB2 V7 subsystem with the following input:

```
DSN SYSTEM(JUDY)

BIND PACKAGE (WSCDBP0.DSNJDBC) MEMBER(DSNJDBC1) ISOLATION(UR)
BIND PACKAGE (WSCDBP0.DSNJDBC) MEMBER(DSNJDBC2) ISOLATION(CS)
BIND PACKAGE (WSCDBP0.DSNJDBC) MEMBER(DSNJDBC3) ISOLATION(RS)
BIND PACKAGE (WSCDBP0.DSNJDBC) MEMBER(DSNJDBC4) ISOLATION(RR)

BIND PLAN(DSNJDBC) ACTION(REPLACE) RETAIN -
    PKLIST(DSNJDBC.DSNJDBC1, -
           DSNJDBC.DSNJDBC2, -
           DSNJDBC.DSNJDBC3, -
           DSNJDBC.DSNJDBC4, -
           WSCDBP0.DSNJDBC.DSNJDBC1, -
           WSCDBP0.DSNJDBC.DSNJDBC2, -
           WSCDBP0.DSNJDBC.DSNJDBC3, -
           WSCDBP0.DSNJDBC.DSNJDBC4)

END
```

This job looks like the one originally a part of the DB2 install job DSNTJJCL in DSN710.SDSNSAMP, but it binds four more packages, using the same DBRMs as the original job (See sample in the Appendix). They are prefixed with WSCDBP0.DSNJDBC, the Location and Collection names respectively for our remote DB2 V6 data sharing group. These are the packages which will be executed at the remote system WSCDBP0, as we invoke the sample from the JUDY ssid. We can submit the bind job from the local JUDY ssid and via the location name, a DB2 member of the WSCDBP0 group will actually perform the bind. It can be any of the four members which are available. See "Considerations for the DB2 V6 Data Sharing Group" for more data sharing techniques which can be used.

The second part of the bind job is to bind the DSNJDBC plan on JUDY. This enables the access to the remote packages. The four packages added to the package list again are the same ones as the original packages, but have the DB2 Location name as the first qualifier.

THE WAS 4.0 DB2 V7 Conundrum

We want to replace the original DSNJDBC plan, which we bound during the JDBC installation process. We do not want to invalidate the permissions which were granted to this plan, as the ACTION(REPLACE) will do on its own. We specified the RETAIN option, which is not the default for BIND PLAN. It allows all previous authorities to be retained.

Because the DB2 V7 DBRMs have "Version" support (a timestamp in the VERSION) column of SYSIBM.SYSPACKAGE, they will not overlay the DB2 V6 (or V5) packages of the same name which are invoked by local DBP* clients. See the sample query in the Appendix, which shows the DSNJDBC* packages.

When the bind job is successful, it also demonstrates that the CDB is configured correctly and communication is actually possible between the V7 DB2 system and one or more members of the DB2 V6 system.

Step 4: Tested JDBC access to wscdbp0 from WSC4 using the DB2 V7 subsystem.

The following program was created to run on the JUDY subsystem running on WSC4 and access data in both the V6 and V7 subsystems. The original code was taken from the DB2 sample program sample01.java and modified to use DB2DataSource. The shell script follows.

```
// NAME = sample01x.java
//
// DESCRIPTIVE NAME = JDBC sample01 application
//
// LICENSED MATERIALS - PROPERTY OF IBM
// 5655-DB2
// (C) COPYRIGHT 1982, 1997 IBM CORP. ALL RIGHTS RESERVED.
//
//
// DB2 JDBC sample01.java application:
//
// (a) Create the DB2 for OS/390 JDBC DataSource
// (b) Create Connection instance
// (c) Create a Statement instance
// (d) Execute a Query and generate a ResultSet instance
// (e) Print column 1 (table name) to system.out
// (f) Close the ResultSet
// (g) Close the Statement
// (h) Close the Connection
//

import java.sql.*;           // JDBC base
import javax.naming.*;      // JNDI Naming Services
import javax.sql.*;         // JDBC 2.0 standard ext. APIs
import com.ibm.db2.jcc.*;   // DB2 implementation of JDBC 2.0
                           // standard extension APIs

public class sample01x {

    public static void main(String args[]) {

        try {

            // Create the connection
            DB2DataSource db2ds = new com.ibm.db2.jcc.DB2DataSource();
            db2ds.setDatabaseName(args[0]);
            db2ds.setDescription("Our Sample Database");
```

THE WAS 4.0 DB2 V7 Conundrum

```
Connection con=db2ds.getConnection();

System.out.println("**** JDBC Connection to DB2 for OS/390.");

// Create the Statement
Statement stmt = con.createStatement();
System.out.println("**** JDBC Statement Created");

// Execute a Query and generate a ResultSet instance
// The Query is a Select from SYSIBM.SYSTABLES
ResultSet rs = stmt.executeQuery("SELECT * FROM SYSIBM.SYSTABLES");
System.out.println("**** JDBC Result Set Created");

// Print all of the table names to sysout
while (rs.next()) {
    String s = rs.getString(1);
    System.out.println("Table NAME = " + s);
}
System.out.println("**** JDBC Result Set output completed");

// Close the resultset
rs.close();
// Close the statement
stmt.close();
System.out.println("**** JDBC Statement Closed");

// Close the connection
con.close();
System.out.println("**** JDBC Disconnect from DB2 for OS/390.");

System.out.println("**** JDBC Exit from class sample01 - no"
Errors.");
    } catch( Exception e ) {
        e.printStackTrace();
    }
}
}
```

The Preparation

A db2sqljjdbc properties file was created during the initial customization of JDBC on the DB2 V7 system. The file resides in the /usr/lpp/db2/db2710/classes directory and is named judy_db2sqljjdbc.properties. Similarly, there are corresponding properties files for each of the DB2 V6 subsystems in the corresponding DB2 V6 directories with names like dbpX_db2sqljjdbc.properties (where X=1,2,3, or 4).

The judy_db2sqljjdbc.properties file contents are found in the Appendix.

DB2 Privileges

Before sample01x.java is run, the user (i.e., mccoxx) must have been granted SELECT authority on SYSIBM.SYSTABLES on both the DB2 V6 system and the DB2 V7 system. The sample selects all rows from SYSIBM.SYSTABLES. See the Appendix for samples of the GRANTS.

The Sample01x Script

THE WAS 4.0 DB2 V7 Conundrum

After performing the javac compile, we show the script (i.e., run_sample01x.sh) used to set up the environment and run the sample01x program:

```
#!/bin/sh
#
# The DB2 V7 libraries are not in the linklist, thus we need a
# STEPLIB
#

export STEPLIB=DSN710.JUDY.SDSNEXIT:DSN710.SDSNLOAD:DSN710.SDSNLOD2

MYPATH=/usr/lpp/db2/db2710/bin:
MYPATH=$MYPATH:/usr/lpp/java2/J1.3/bin:
MYPATH=$MYPATH:/usr/lpp/java2/J1.3/bin/classic:
MYPATH=$MYPATH:/usr:/usr/sbin
export PATH=$MYPATH

export LIBPATH=/usr/lpp/db2/db2710/lib:/lib:/usr/lib

export LD_LIBRARY_PATH=/usr/lpp/db2/db2710/lib

MYCLASSPATH=./usr/lpp/db2/db2710/classes/db2j2classes.zip:
MYCLASSPATH=$MYCLASSPATH:/usr/lpp/WebSphere/lib/ws390crt.jar
export CLASSPATH=$MYCLASSPATH

export DB2SQLJPROPERTIES=/usr/lpp/db2/db2710/classes/judy_db2sqljjdbc.properties

echo "Location: " $1
java sample01x $1
```

In our DB2 V7 environment (JUDY), we ran the script specifying:

```
> run_sample01x.sh wscjudy
- and -
> run_sample01x.sh wscdbp0
```

A successful run of both indicates that JDBC access to the 'local' and 'remote' system does indeed work.

The Output

We directed the output to wscdbp0.txt. Due to the volume, only a portion of the results of the SYSIBM.SYSTABLES query is shown.

```
Location: wscdbp0
**** JDBC Entry within class sample01.
**** JDBC Entry URL: jdbc:db2os390sqlj:wscdbp0
**** JDBC Connection to DB2 for OS/390.
**** JDBC Statement Created
**** JDBC Result Set Created
Table NAME = SYSCOPY
Table NAME = SYSCOLAUTH
Table NAME = SYSCOLUMNS
. . .
Table NAME = MVSPM_ENQUEUE_HV
Table NAME = MVSPM_LPAR_H
**** JDBC Result Set output completed
**** JDBC Statement Closed
**** JDBC Disconnect from DB2 for OS/390.
```


THE WAS 4.0 DB2 V7 Conundrum

**** JDBC Exit from class sample01 - no Errors.

DB2 V6 Local JDBC Execution

Despite our having bound the DB2 V7 DSNJDBC* packages on WSCDBP0, our DBP* local users can continue to execute their existing JDBC applications. Their packages have not been changed, nor has their local DSNJDBC plan. A common problem occurs when packages of the same name are bound on a DB2 subsystem. The user receives an SQLCODE -805, which normally means that a package has been replaced but the executable load module (which contains the timestamp for each SQL statement call) has not, and the incompatibility is disallowed by DB2. JDBC does not have a load module. Where does the timestamp mismatch occur? The DSNJDBC_JDBCProfile.ser file (.ser for short) can be considered the executable, and it contains the timestamp. As long as the .ser file from DB2 V6 remains in the CLASSPATH, the DB2 V6 JDBC packages will continue to run. The reason our remote bind job did not overlay those from DB2 V6, which have the same name, is that DB2 V7 JDBC packages have “versioning”.

The sample query in the Appendix demonstrates that some of the packages have a version id and some do not. Those which do not are for DB2 V6, at this writing.

You cannot execute sample01x on DB2 V6, because the program invokes the Java 2.0 DataSource method, which requires the DB2 V7 JDBC2 driver. The DB2 V6 JDBC driver does not support the JAVA 2.0 specification.

The DB2 V7 JDBC2 driver will not be retrofitted to either DB2 V5 or DB2 V6. DB2 V7 also ships a JDBC driver which implements the JDBC 1.2 specification. The latter driver is supplied for compatibility only and cannot be used with WAS 4.0.

Considerations for a DB2 V6 Data Sharing Group

The essence of DB2 data sharing is that there is one copy of the data and one DB2 Catalog and Directory. All DB2 subsystems which are members of the group share the data. In our environment, requests can be routed to any member which is active: DBP1, DBP2, DBP3, DBP4.

To verify that you can route the sample to any member, stop all other members, and run the test. To execute the sample using DBP2, for example, we stopped DBP4, DBP3, and DBP1.

Excluding a DB2 member as a Server

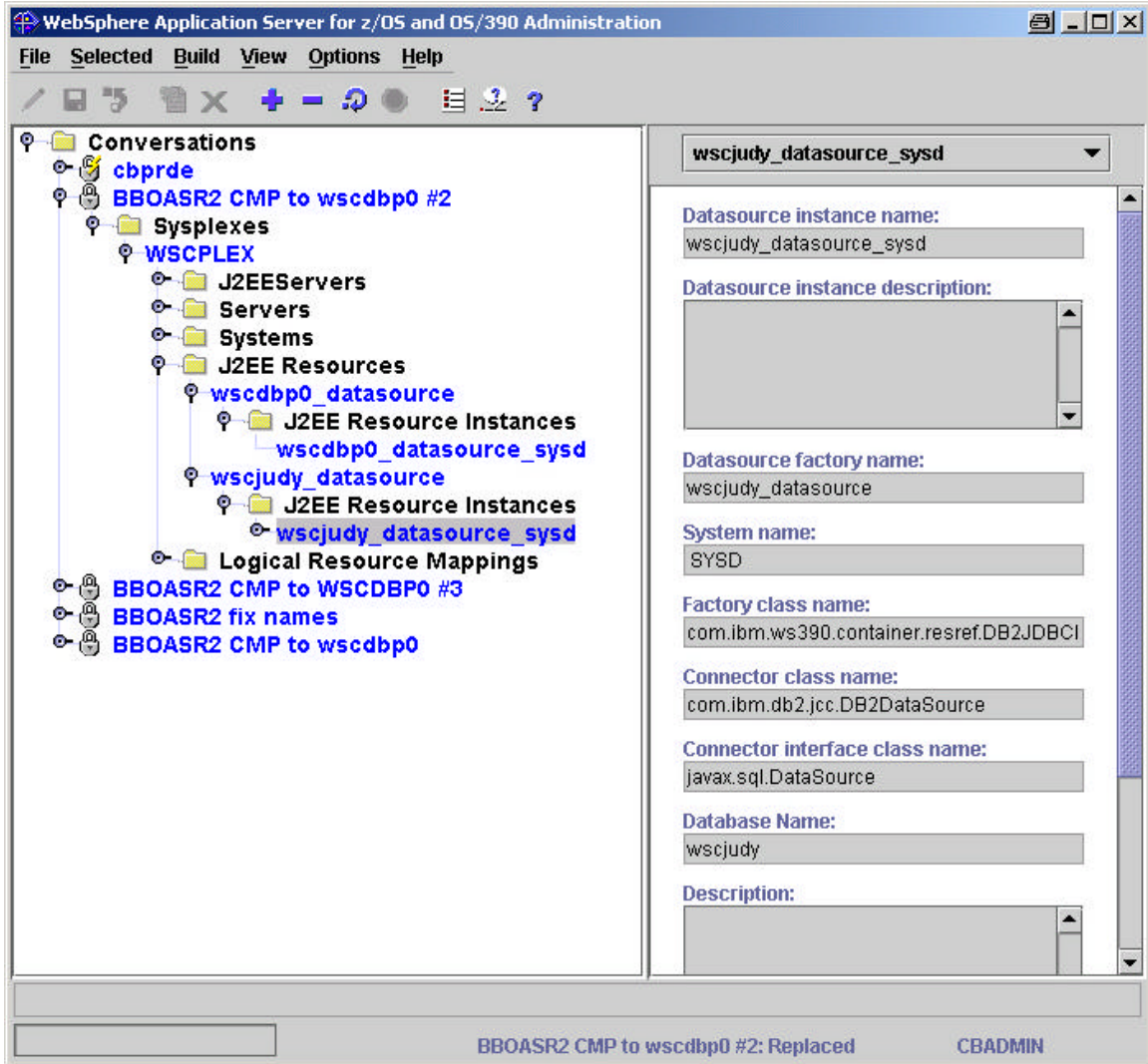
You can permanently exclude one of the members in the group as a server by changing a value in the member's DSNZPARM. To exclude a member from DDF server processing, set the MAX REMOTE ACTIVE option of installation panel DSNTIPE to zero for that member (or job SDSNSAMP(DSNTIJUZ) macro keyword MAXDBAT). Submit the resulting DSNTIJUZ job. The effects of setting this parameter to zero are:

- ⚠⚠ DB2 does not register the member's LU name with the VTAM generic LU name during DDF startup. Connections that use the generic LU name are directed to those members for which MAX REMOTE ACTIVE is greater than zero.
- ⚠⚠ DB2 does not register the member to WLM for member routing. Its name is not included in the list of available LUs that is sent to remote sites. Therefore, DDF server work is never routed to that member.
- ⚠⚠ DB2 does not listen on the DRDA SQL port, which means that SQL TCP/IP requests are accepted only by members for which MAXDBAT is greater than zero.

THE WAS 4.0 DB2 V7 Conundrum

Step 5: Configured DataSources in WAS for EJBs to use WSCDBP0

In the deployment of BMP and CMP beans backed with DB2 data, there will be a resource reference for a DB2 data source which must be resolved. Basically, the process involves defining a Data Source and a Data Source Instance for each of the DB2 systems to which one wants to connect. On the WSC WAS 4.0 system we have data sources for DB2 Locations 'wscjudy' and 'wscdbp0'. Here is the view from the Systems Management Administrative tool:

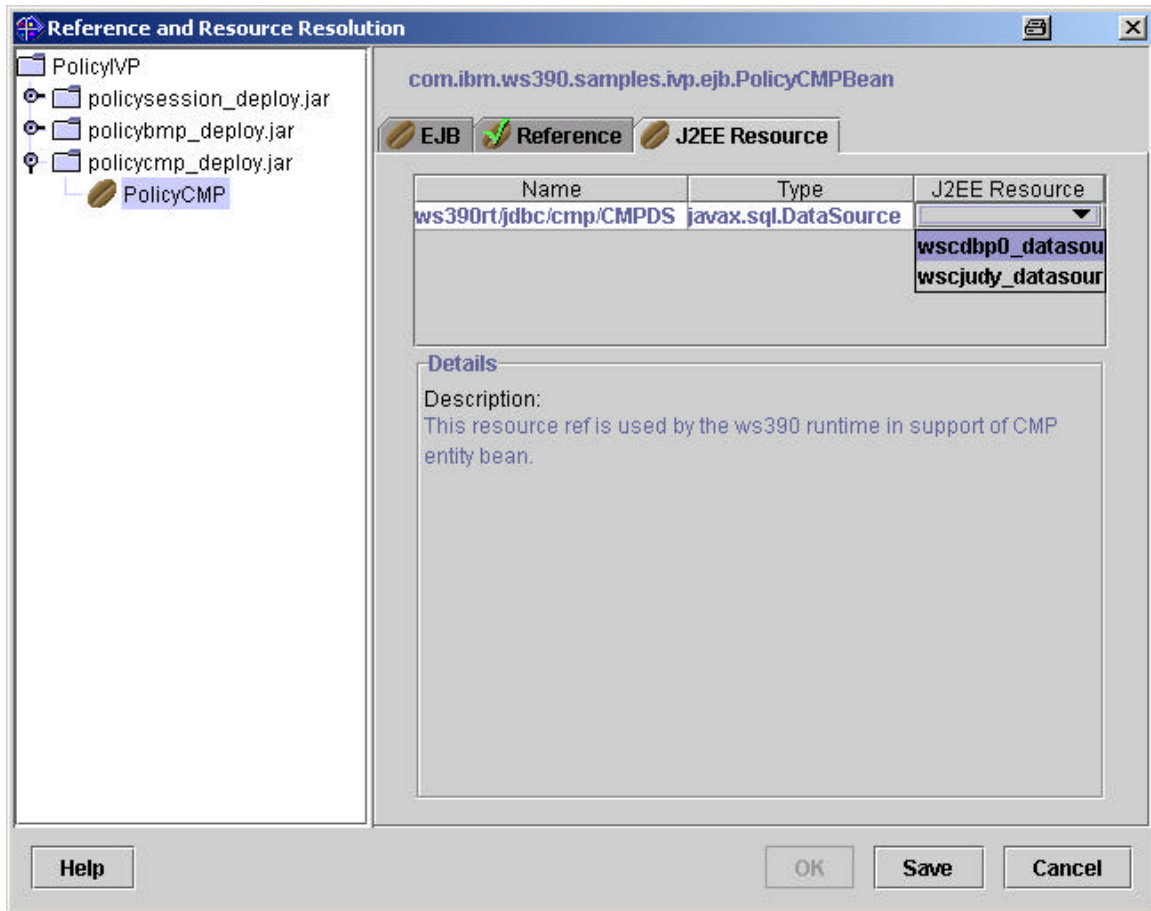


The above figure illustrates the interesting portion of the datasources for 'wscjudy'; there is a corresponding entry for 'wscdbp0'. These data sources and data source instances were created once and can be used by any or all EJBs which need data from one or the other DB2 systems.

Validation

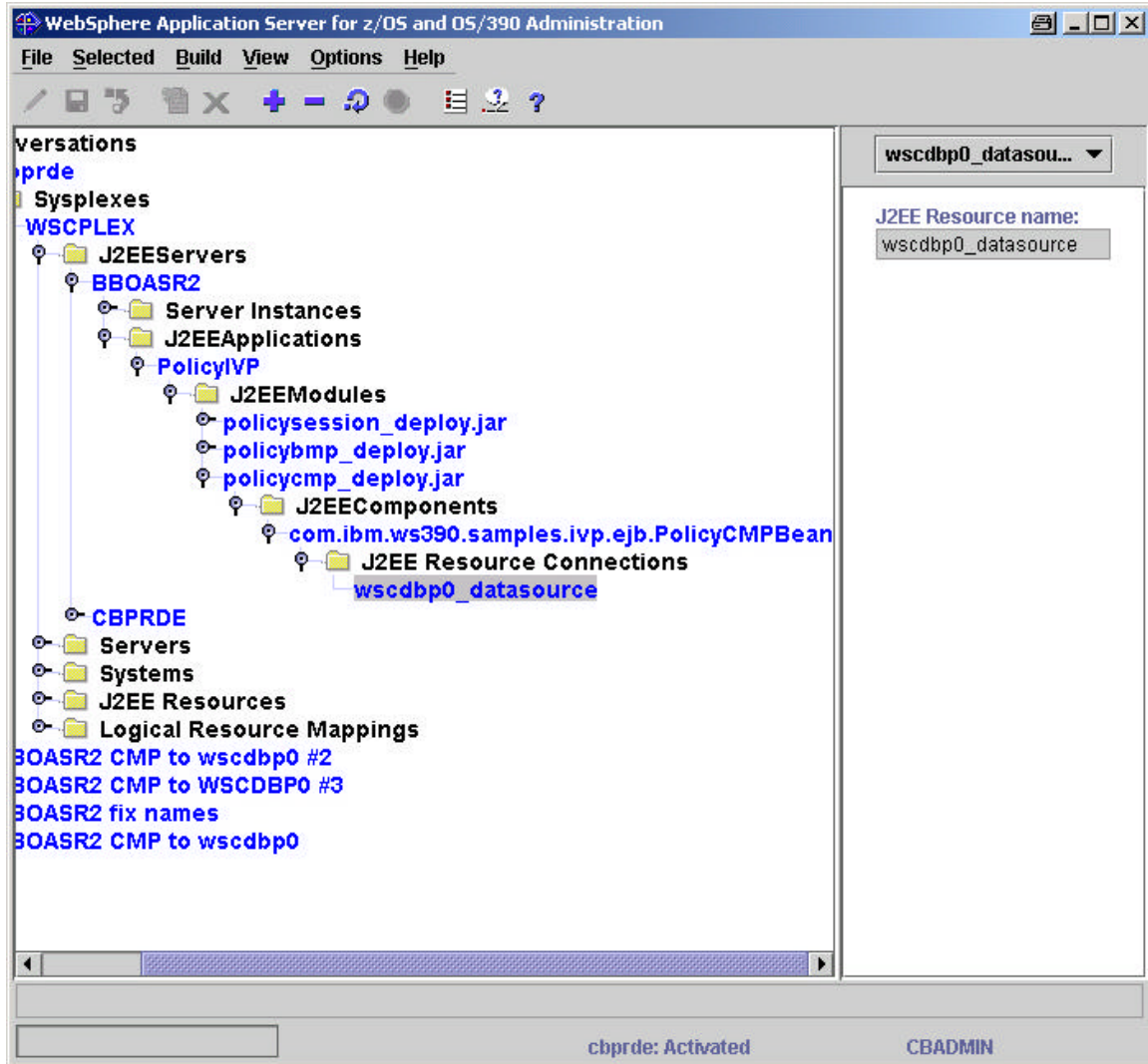
The Policy IVP application contains a CMP and BMP EJB. During the installation of this application into a server, one is forced to resolve the resource reference to a DB2 datasource. In this instance, we want the CMP bean to reference data resident in the DB2 V6 system (wscdbp0) and the BMP bean to reference data in the DB2 V7 system (wscjudy). For example, you will see the following panel where the CMP datasource reference was associated with the wscdbp0_datasource previously defined in the WAS configuration.

THE WAS 4.0 DB2 V7 Conundrum



Once the resource reference is resolved and the deployment process is complete, you can see this bean is indeed backed by data in the DB2 Version 6 data sharing group.

THE WAS 4.0 DB2 V7 Conundrum



Now it is a simple matter of creating the table (i.e., BBO.POLICYDO) referenced in both wscjudy and wscdbp0 locations, grant permission for the userid associated with the server to Select, Update, Insert, and Delete from the BBO.POLICYDO table and run the IVP script. The 'bmp' EJB bean test will update the table in wscjudy and the 'cmp' EJB bean test will update the table in wscdbp0.

For example, running the script from the shell, one sees the following:

```
:/usr/lpp/WebSphere/samples/PolicyIVP/ejb-> ejbivp.sh
***** bmp bean will be run!
Look up policy session home
Get property file name from system properties
Load property file properties into a property object
com.ibm.ws.naming.ldap.WsnLdapInitialContextFactory
Create initial context

Lookup policy home
Narrow home
Starting tran 1
```

THE WAS 4.0 DB2 V7 Conundrum

```
Starting tran 2
bmp IVP has completed successfully
***** cmp bean will be run!
Look up policy session home
Get property file name from system properties
Load property file properties into a property object
com.ibm.ws.naming.ldap.WsnLdapInitialContextFactory
Create initial context

Lookup policy home
Narrow home
Starting tran 1
Starting tran 2
cmp IVP has completed successfully
:/usr/lpp/WebSphere/samples/PolicyIVP/ejb->
```

Now, to verify this actually worked, logon to TSO on the WSC4 system with a userid having the authority to select from BBO.POLICYDO and run a spufi script to select everything in the table at each location, you will see the following:

```
-----+-----+-----+-----+-----+-----+-----
select * from wscjudy.bbo.policydo;
-----+-----+-----+-----+-----+-----+-----
                PAMOUNT      PNUMBER      PPREMIUM
-----+-----+-----+-----+-----+-----+-----
+0.11000000000000000E+02      1111      +0.15000000000000000E+03
+0.11000000000000000E+02      7777      +0.38800000000000000E+03
DSNE610I NUMBER OF ROWS DISPLAYED IS 2
DSNE616I STATEMENT EXECUTION WAS SUCCESSFUL, SQLCODE IS 100
-----+-----+-----+-----+-----+-----+-----
select * from wscdbp0.bbo.policydo;
-----+-----+-----+-----+-----+-----+-----
                PAMOUNT      PNUMBER      PPREMIUM
-----+-----+-----+-----+-----+-----+-----
+0.13200000000000000E+03      2222      +0.35000000000000000E+03
+0.13200000000000000E+03      8888      +0.67600000000000000E+03
DSNE610I NUMBER OF ROWS DISPLAYED IS 2
DSNE616I STATEMENT EXECUTION WAS SUCCESSFUL, SQLCODE IS 100
-----+-----+-----+-----+-----+-----+-----
```

Summary:

The WSC was successful in implementing WAS 4.0 using DB2 V7 for only the WAS runtime tables and was able to deploy EJBs backed by DB2 in a 'remote' DB V6 system. We hope the discussion of the WSC environment and actions taken to accomplish this goal will assist others and reduce the time spent on the learning curve. The bottom line is the environment setup is not difficult and it works.

As always, your mileage may vary and always consult the product publications.

THE WAS 4.0 DB2 V7 Conundrum

Appendix: JDBC Sample Setup

Below we have listed the jobs and their output as additional information.

- ⚔ Set Up the CDB on DB2 V7 (JUDY) and Query Results
- ⚔ Properties file – judy_db2sqljdbc.properties
- ⚔ Grant Execute on SYSTABLES to users on DB2 V7
- ⚔ Grant Execute on SYSTABLES to users on DB2 V6
- ⚔ Original DSNJDBC Bind distributed with DB2 V7
- ⚔ Original GRANT on DSNJDBC distributed with DB2 V7
- ⚔ A Sample SPUFI Query run from the DB2 V7 JUDY system

Set Up the CDB on DB2 V7 (JUDY) and Query Results

```
//jobname JOB 1
//JOBLIB DD DSN=DSN710.SDSNEXIT,DISP=SHR
// DD DSN=DSN710.SDSNLOAD,DISP=SHR
//STEP1 EXEC PGM=IKJEFT01,DYNAMNBR=20
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
DSN SYSTEM(JUDY)
RUN PROGRAM(DSNTEP2) PLAN(DSNTEP71) -
LIB('DB2710.JUDY.RUNLIB.LOAD')
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN DD *
INSERT INTO SYSIBM.LOCATIONS (LOCATION, LINKNAME, PORT)
VALUES ('WSCDBP0','WSCDBP0', '446');
INSERT INTO SYSIBM.IPNAMES (LINKNAME, SECURITY_OUT, USERNAMES, IPADDR )
VALUES ('WSCDBP0', 'R', ' ', 'WSCDBP0.WSCPLEX.WASHINGTON.IBM.COM');
SELECT * FROM SYSIBM.LOCATIONS;
SELECT * FROM SYSIBM.IPNAMES;
/*
```

Results of Query

```
-----+-----+-----+-----+-----+-----+-----+
SELECT * FROM SYSIBM.LOCATIONS;
-----+-----+-----+-----+-----+-----+-----+
LOCATION          LINKNAME  IBMREQD  PORT          TPN
-----+-----+-----+-----+-----+-----+-----+
WSCDBP0         WSCDBP0   N         446
DSNE610I NUMBER OF ROWS DISPLAYED IS 1
DSNE616I STATEMENT EXECUTION WAS SUCCESSFUL, SQLCODE IS 100
-----+-----+-----+-----+-----+-----+-----+
SELECT * FROM SYSIBM.IPNAMES;
-----+-----+-----+-----+-----+-----+-----+
LINKNAME  SECURITY_OUT  USERNAMES  IBMREQD  IPADDR
-----+-----+-----+-----+-----+-----+-----+
WSCDBP0   R              N          WSCDBP0.WSCPLEX.WASHINGTON.IBM.COM
DSNE610I NUMBER OF ROWS DISPLAYED IS 1
DSNE616I STATEMENT EXECUTION WAS SUCCESSFUL, SQLCODE IS 100
-----+-----+-----+-----+-----+-----+-----+
```

THE WAS 4.0 DB2 V7 Conundrum

Properties file – judy_db2sqljjdbc.properties

```
#
# This db2jdbcsqlj.properties file is for the JUDY subsystem in support
# of WAS 4.0
# This DBRMLIB is specific for the JUDY subsystem and the JDBC driver.
# Used by the db2genJDBC process, now remains only for documentation
# purposes.
#
#DB2SQLJDBRMLIB=DSN710.JUDY.JDBC.DBRMLIB
#
# This is installation specific:  the name of the DB2 subsystem to use.
#
DB2SQLJSSID=JUDY
#
# The next 5 items are default values which are coded for d
# documentation reasons only.
#
DB2SQLJPLANNAME=DSNJDBC
DB2SQLJJDBCPROGRAM=DSNJDBC
DB2SQLJMULTICONTEXT=YES
DB2SQLJATTACHTYPE=RRSAF
DB2CURSORHOLD=YES
#
# The following items are commented out, they are for tracing,
# there are no default values. And tracing is turned OFF by
# commenting out these items.
#
#DB2SQLJ_TRACE_FILENAME=/tmp/jdbctrace
#DB2SQLJ_TRACE_BUFFSIZE=256
#DB2SQLJ_TRACE_WRAP=1
```

Grant Execute on SYSTABLES to users on DB2 V7

The SYSAFF statement is required because the job must execute on the system where JUDY runs.

```
//jobname JOB 1
/*-----*
/*JOBPARM SYSAFF=WSC4
// SET DB2='DSN710'
//BBOIGR3 EXEC PGM=IKJEFT01,DYNAMNBR=20
//STEPLIB DD DSN=&DB2..SDSNLOAD,DISP=SHR
//DBRMLIB DD DSN=&DB2..SDSNDBRM,DISP=SHR
//SYSTSPRT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSTSIN DD *
DSN SYSTEM(JUDY)
RUN PROGRAM(DSNTIAD) PLAN(DSNTIA71) -
LIB('DB2710.JUDY.RUNLIB.LOAD')
```

THE WAS 4.0 DB2 V7 Conundrum

```
END
//SYSIN DD *
  GRANT SELECT ON TABLE SYSIBM.SYSTABLES TO MCCOX;
/*
```

Grant Execute on SYSTABLES to users on DB2 V6

No SYSAFF statement is required because all our system images have one member of the data sharing group.

We have also specified the group attach, DBP0, on the DSN SYSTEM statement to allow the job to connect to any member of the data sharing group in a non-specific request..

Unlike BIND, GRANT must be executed locally for the system on which privileges are desired

```
/*-----*
//  SET DB2='DSN610'
//BBOIGR3 EXEC PGM=IKJEFT01,DYNAMNBR=20
//STEPLIB DD DSN=&DB2..SDSNLOAD,DISP=SHR
//DBRMLIB DD DSN=&DB2..SDSNDBRM,DISP=SHR
//SYSTSPRT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSTSIN DD *
  DSN SYSTEM(DBP0)
  RUN PROGRAM(DSNTIAD) PLAN(DSNTIA61) -
    LIB('WSCDBP0.DBP0.RUNLIB.LOAD')
END
//SYSIN DD *
  GRANT SELECT ON TABLE SYSIBM.SYSTABLES TO MCCOX;
/*
```

Original DSNJDBC Bind distributed with DB2 V7

The SYSAFF statement is required so the job will execute on WSC4, on which JUDY runs

```
//jobname JOB 1
/*JOBPARM SYSAFF=WSC4
/*-----*
//*
//* BIND THE DSNJDBC PLAN ON THE VERSION 7 SUBSYSTEM.
//*
//*
//*-----*
//BIND EXEC PGM=IKJEFT01,REGION=0M
//STEPLIB DD DISP=SHR,DSN=DSN710.SDSNLOAD
// DD DISP=SHR,DSN=DSN710.JUDY.SDSNEXIT
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//DBRMLIB DD DISP=SHR,DSN=DSN710.JUDY.JDBC.DBRMLIB
//SYSTSIN DD *
  DSN SYSTEM(JUDY)

  BIND PACKAGE (DSNJDBC) MEMBER(DSNJDBC1) ISOLATION(UR)
```


THE WAS 4.0 DB2 V7 Conundrum

```

BIND PACKAGE (DSNJDBC) MEMBER(DSNJDBC2) ISOLATION(CS)
BIND PACKAGE (DSNJDBC) MEMBER(DSNJDBC3) ISOLATION(RS)
BIND PACKAGE (DSNJDBC) MEMBER(DSNJDBC4) ISOLATION(RR)
BIND PLAN(DSNJDBC) -
            PKLIST(DSNJDBC.DSNJDBC1,      -
                  DSNJDBC.DSNJDBC2,      -
                  DSNJDBC.DSNJDBC3,      -
                  DSNJDBC.DSNJDBC4)

      END
/*

```

Original GRANT on DSNJDBC for DB2 V7

The SYSAFF statement is required so the job will execute on WSC4, on which JUDY runs

```

//jobname JOB 1
/*JOBPARM SYSAFF=WSC4
//  SET DB2='DSN710'
//BBOIGR3 EXEC PGM=IKJEFT01,DYNAMNBR=20
//STEPLIB DD DSN=&DB2..SDSNLOAD,DISP=SHR
//DBRMLIB DD DSN=&DB2..SDSNDBRM,DISP=SHR
//SYSTSPRT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSTSIN DD *
DSN SYSTEM(JUDY)
RUN PROGRAM(DSNTIAD) PLAN(DSNTIA71) -
      LIB('DB2710.JUDY.RUNLIB.LOAD')
END
//SYSIN DD *
      GRANT EXECUTE ON PLAN DSNJDBC           TO PUBLIC;
      GRANT EXECUTE ON PACKAGE DSNJDBC.*      TO PUBLIC;
/*

```

A Sample SPUFI Query run from the DB2 V7 JUDY system:

This was run on SPUFI from wscjudy.

This is the input (we didn't do Select * because we were trying minimize the line width)

```

SELECT * FROM WSCDBP0.SYSIBM.SYSPLAN        WHERE NAME LIKE 'DSNJDBC%';

SELECT LOCATION, COLLID, NAME, HEX(CONTOKEN), BINDTIME, VERSION
      FROM WSCDBP0.SYSIBM.SYSPACKAGE  WHERE NAME LIKE 'DSNJDBC%';

```

This is the output. To get a display from SYSPACKAGE on one line, we took artistic license and do not show LOCATION in the query, as it was blank when issued from the DBP1 ssid.

```

-----+-----+-----+-----+-----+-----+-----+-----+-----
--SELECT FROM V6 SYSTEM
      SELECT * FROM WSCDBP0.SYSIBM.SYSPLAN        WHERE NAME LIKE 'DSNJDBC%';

```

THE WAS 4.0 DB2 V7 Conundrum

```

-----+-----+-----+-----+-----+-----+-----+-----
NAME      CREATOR  BINDDATE  VALIDATE  ISOLATION  VALID  OPERATIVE  BINDTIME
-----+-----+-----+-----+-----+-----+-----+-----
DSNJDBC   SYSADM1  010319   R         R          Y     Y         173925652232

```

```

DSNE610I NUMBER OF ROWS DISPLAYED IS 1      DSNE616I STATEMENT EXECUTION WAS SUCCESSFUL,
SQLCODE IS 100

```

```

-----+-----+-----+-----+-----+-----+-----+-----
      SELECT  LOCATION, COLLID, NAME, HEX(CONTOKEN), BINDTIME, VERSION
      FROM    WSCDBP0.SYSIBM.SYSPACKAGE WHERE NAME LIKE 'DSNJDBC%';

```

```

-----+-----+-----+-----+-----+-----+-----+-----
..COLLID  NAME      CONTOKEN      BINDTIME      VERSION
-----+-----+-----+-----+-----+-----+-----+-----
DSNJDBC   DSNJDBC1  000000E5587B99A7  2001-03-19-17.39.24.581035
DSNJDBC   DSNJDBC1  000000E55872C081  2001-03-21-18.16.49.216256  2001-3-19-14.50.26.177
DSNJDBC   DSNJDBC2  000000E5587B99A7  2001-03-19-17.39.24.863606
DSNJDBC   DSNJDBC2  000000E55872C081  2001-03-21-18.02.33.355657  2001-3-19-14.50.26.177
DSNJDBC   DSNJDBC3  000000E5587B99A7  2001-03-19-17.39.25.146399
DSNJDBC   DSNJDBC3  000000E55872C081  2001-03-21-18.16.47.984016  2001-3-19-14.50.26.177
DSNJDBC   DSNJDBC4  000000E5587B99A7  2001-03-19-17.39.25.402972
DSNJDBC   DSNJDBC4  000000E55872C081  2001-03-21-18.16.48.564477  2001-3-19-14.50.26.177

```

```

DSNE610I NUMBER OF ROWS DISPLAYED IS 8
DSNE616I STATEMENT EXECUTION WAS SUCCESSFUL, SQLCODE IS 100

```

Observe that there are two sets of packages. One set has a "Version "; those are the DB2 V7 packages which were remotely bound to the WSCDBP0 location. The others are for DB2 V6, generated from the JDBC installation procedure for DB2 V6. Versioning support is not yet available for DB2 V6/V5 JDBC. They actually carry a 'NULL' version. Thus the DB2 V7 package bind did not overlay those from DB2 V6. They are still available for local execution as long as their .ser file is in the CLASSPATH.

The End